View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Open Archive Toulouse Archive Ouverte

Open Archive TOULOUSE Archive Ouverte (OATAO)

OATAO is an open access repository that collects the work of Toulouse researchers and makes it freely available over the web where possible.

This is an author-deposited version published in : http://oatao.univ-toulouse.fr/
Eprints ID : 12808

Official URL: http://dx.doi.org/10.1007/978-3-319-11683-9_3

To cite this version : Vanaret, Charlie and Gotteland, Jean-Baptiste and Durand, Nicolas and Alliot, Jean-Marc *Preventing Premature Convergence and Proving the Optimality in Evolutionary Algorithms.* (2013) In: 11th International Conference Evolution Artificielle (EA 2013), 21 October 2013 - 23 October 2013 (Bordeaux, France).

Any correspondance concerning this service should be sent to the repository administrator: staff-oatao@listes-diff.inp-toulouse.fr

Preventing Premature Convergence and Proving the Optimality in Evolutionary Algorithms

Charlie Vanaret[1,2](✉), Jean-Baptiste Gotteland[1,2], Nicolas Durand[1,2], and Jean-Marc Alliot[2]

[1] Laboratoire de Mathématiques Appliquées, Informatique et Automatique pour l'Aérien, Ecole Nationale de l'Aviation Civile, Toulouse, France
{vanaret,gottelan,durand}@recherche.enac.fr
[2] Institut de Recherche en Informatique de Toulouse, Toulouse, France
jean-marc.alliot@irit.fr

Abstract. Evolutionary Algorithms (EA) usually carry out an efficient exploration of the search-space, but get often trapped in local minima and do not prove the optimality of the solution. Interval-based techniques, on the other hand, yield a numerical proof of optimality of the solution. However, they may fail to converge within a reasonable time due to their exponential complexity and their inability to quickly compute a good approximation of the global minimum. The contribution of this paper is a hybrid algorithm called Charibde in which a particular EA, Differential Evolution, cooperates with a branch and bound algorithm endowed with interval propagation techniques. It prevents premature convergence toward local optima and is highly competitive with both deterministic and stochastic existing approaches. We demonstrate its efficiency on a benchmark of highly multimodal problems, for which we provide previously unknown global minima and certification of optimality.

1 Motivation

Evolutionary Algorithms (EA) have been widely used by the global optimization community for their ability to handle complex problems with no assumption on continuity or differentiability. They generally converge toward satisfactory solutions, but may get trapped in local optima and provide suboptimal solutions. Moreover, their convergence remains hard to control due to their stochastic nature. On the other hand, exhaustive Branch and Bound methods based on Interval Analysis [1] guarantee rigorous bounds on the solutions to numerical optimization problems but are limited by their exponential complexity.

Few methods attempted to hybridize EA and branch and bound algorithms in which lower bounds of the objective function are computed using Interval Analysis. The approaches in the literature are essentially *integrative*, in that they embed one algorithm within the other. Sotiropoulos et al. [2] used an Interval Branch and Bound (IB&B) to reduce the domain to a list of ε-large subspaces. A Genetic Algorithm (GA) [3] was then initialized within each subspace to improve the upper bound of the global minimum. Zhang and Liu [4] and Lei and Chen [5] used respectively a GA and mind evolutionary computation within the IB&B to improve the bounds and the exploration of the remaining subspaces. In a previous communication [6], we proposed a *cooperative* approach combining the efficiency of a GA and the reliability of Interval Analysis. We presented new optimality results for two multimodal benchmark functions (Michalewicz, dimension 12 and rotated Griewank, dimension 8), demonstrating the validity of the approach. However, techniques that exploit the analytical form of the objective function, such as local monotonicity and constraint programming, were not addressed. In this paper, we propose an advanced cooperative algorithm, Charibde (Cooperative Hybrid Algorithm using Reliable Interval-Based methods and Differential Evolution), in which a Differential Evolution algorithm cooperates with interval propagation methods. New optimal results achieved on a benchmark of difficult multimodal functions attest the substantial gain in performance.

The rest of the paper is organized as follows. Notations of Interval Analysis are introduced in Sect. 2 and interval-based techniques are presented in Sect. 3. The implementation of Charibde is detailed in Sect. 4. Results on a benchmark of test functions are discussed in Sect. 5.

2 Interval Analysis

Interval Analysis (IA) bounds round-off errors due to the use of floating-point arithmetic by computing interval operations with outward rounding [1]. Interval arithmetic extends real-valued functions to intervals.

Definition 1 (Notations). *An interval* $X = [\underline{X}, \overline{X}]$ *with floating-point bounds defines the set* $\{x \in \mathbb{R} \mid \underline{X} \leq x \leq \overline{X}\}$. $\mathbb{IR}$ *denotes the set of real intervals. We note* $m(X) = \frac{1}{2}(\underline{X} + \overline{X})$ *its midpoint. A box* $\mathbf{X} = (X_1, \ldots, X_n)$ *is an interval vector. We note* $m(\mathbf{X}) = (m(X_1), \ldots, m(X_n))$ *its midpoint. We note* $\square(X, Y)$ *the convex hull of two boxes* X *and* Y*, that is the smallest box that contains* X *and* Y*.*

In the following, capital letters represent interval quantities (interval X*) and bold letters represent vectors (box* $\mathbf{X}$*, vector* $\mathbf{x}$*).*

Definition 2 (Interval extension; Natural interval extension). *Let* $f : \mathbb{R}^n \to \mathbb{R}$ *be a real-valued function.* $F : \mathbb{IR}^n \to \mathbb{IR}$ *is an interval extension of* f *if*

$$\forall \mathbf{X} \in \mathbb{IR}^n, f(\mathbf{X}) = \{f(\mathbf{x}) \mid \mathbf{x} \in \mathbf{X}\} \subset F(\mathbf{X})$$
$$\forall (\mathbf{X}, \mathbf{Y}) \in \mathbb{IR}^n, \mathbf{X} \subset \mathbf{Y} \Rightarrow F(\mathbf{X}) \subset F(\mathbf{Y})$$

The natural interval extension F_N *is obtained by replacing the variables with their domains and real elementary operations with interval arithmetic operations.*

The quality of enclosure of $f(X)$ depends on the syntactic form of f: the natural interval extensions of different but equivalent expressions may yield different ranges (Example 1). In particular, IA generally computes a large overestimation of the image due to multiple occurrences of a same variable, considered as different variables. This **dependency problem** is the main source of overestimation when using interval computations. However, an appropriate rewriting of the expression may reduce or overcome dependency: if f is continuous inside a box, its natural interval extension F_N yields the optimal image when each variable occurs only once in its expression.

Example 1. Let $f(x) = x^2 - 2x, g(x) = x(x-2)$ and $h(x) = (x-1)^2 - 1$, where $x \in X = [1, 4]$. f, g and h have equivalent expressions, however computing their natural interval extensions yields

$$\begin{aligned}
F_N([1,4]) &= [1,4]^2 - 2 \times [1,4] = [1,16] - [2,8] = [-7,14] \\
G_N([1,4]) &= [1,4] \times ([1,4] - 2) = [1,4] \times [-1,2] = [-4,8] \\
H_N([1,4]) &= ([1,4] - 1)^2 - 1 = [0,3]^2 - 1 = [0,9] - 1 = [-1,8]
\end{aligned}$$

We have $f([1,4]) = H_N([1,4]) \subset G_N([1,4]) \subset F_N([1,4])$.

3 Interval-Based Techniques

Interval Branch and Bound Algorithms (IB&B) exploit the conservative properties of interval extensions to rigorously bound global optima of numerical optimization problems [7]. The method consists in splitting the initial search-space into subspaces (branching) on which an interval extension F of the objective function f is evaluated (bounding). By keeping track of the best upper bound $\tilde{f}$ of the global minimum f^*, boxes that certainly do not contain a global minimizer are discarded (Example 2). The remaining boxes are stored to be processed at a later stage until the desired precision ε is reached. The process is repeated until all boxes have been processed. Convergence certifies that $\tilde{f} - f^* < \varepsilon$, even in the presence of rounding errors. However, the exponential complexity of IB&B hinders the speed of convergence on large problems.

Example 2. Let us detail the first step of the IB&B on the problem $\min_{x \in X} f(x) = x^4 - 4x^2$ over the interval $X = [-1, 4]$. The natural interval extension of f is $F_N(X) = X^4 - 4X^2$ and $F_N([-1,4]) = [-64, 256] \supset [-4, 192] = f([-1,4])$. The floating-point evaluation $f(1) = -3$ yields an upper bound $\tilde{f}$ of f^*. Evaluating F_N on the subinterval $[3, 4]$ reduces the overestimation induced by the dependency effect: $F_N([3,4]) = [17, 220] \supset [45, 192] = f([3,4])$. Since this enclosure is rigorous, $\forall x \in [3,4], f(x) \geq 17 > \tilde{f} = -3 \geq f^*$. Therefore, the interval $[3, 4]$ cannot contain a global minimizer and can be safely discarded.

Interval Constraint Programming (ICP) aims at solving systems of nonlinear equations and numerical optimization problems. Stemming from Interval Analysis and Interval Constraint Programming communities, filtering/contraction algorithms [8] narrow the bounds of the variables without loss of solutions.

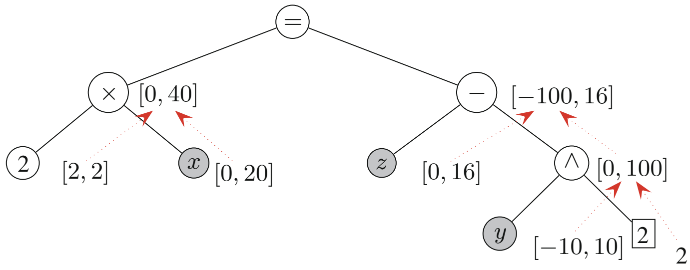

Fig. 1. `HC4Revise`: evaluation phase

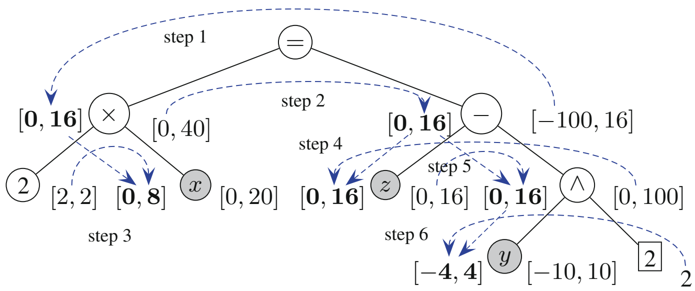

Fig. 2. `HC4Revise`: propagation phase

The standard contraction algorithm `HC4Revise` [9] carries out a double exploration of the syntax tree of a constraint to contract each occurrence of the variables (Example 3). It consists in an evaluation (bottom-up) phase that computes the elementary operation of each node, and a backward (top-down) propagation phase using inverse functions.

Example 3. Let $2x = z - y^2$ be an equality constraint, with $x \in [0, 20]$, $y \in [-10, 10]$ and $z \in [0, 16]$. The elementary expressions are the nodes $n_1 = 2x$, $n_2 = y^2$ and $n_3 = z - n_2$.

The evaluation phase (Fig. 1) computes $n_1 = 2 \times [0, 20] = [0, 40]$, $n_2 = [-10, 10]^2 = [0, 100]$ and $n_3 = [0, 16] - [0, 100] = [-100, 16]$.

The propagation phase (Fig. 2) starts by intersecting n_1 and n_3 (steps 1 and 2), then computes the inversion of each elementary expression (steps 3 to 6).

- steps 1 and 2: $n_1' = n_3' = n_1 \cap n_3 = [0, 40] \cap [-100, 16] = [0, 16]$
- step 3: $x' = x \cap \frac{n_1'}{2} = [0, 20] \cap [0, 8] = [0, 8]$
- step 4: $z' = z \cap (n_2 + n_3') = [0, 16] \cap ([0, 100] + [0, 16]) = [0, 16]$
- step 5: $n_2' = n_2 \cap (z' - n_3') = [0, 100] \cap ([0, 16] - [0, 16]) = [0, 16]$
- step 6: $y' = \Box(y \cap -\sqrt{n_2'}, y \cap \sqrt{n_2'}) = \Box([-4, 0], [0, 4]) = [-4, 4]$

The initial box $([0, 20], [-10, 10], [0, 16])$ has been reduced to $([0, 8], [-4, 4], [0, 16])$ without loss of solutions.

When partial derivatives are available, detecting **local monotonicity** with respect to a variable cancels the dependency effect due to this variable (Definition 3 and Example 4). In Definition 3, we call a monotonic variable a variable with respect to which f is monotonic.

Definition 3 (Monotonicity-based extension). *Let f be a function involving the set of variables $\mathcal{V}$. Let $\mathcal{X} \subseteq \mathcal{V}$ be a subset of k monotonic variables and $\mathcal{W} = \mathcal{V} \backslash \mathcal{X}$ the set of variables not detected monotonic. If x_i is an increasing (resp. decreasing) variable, we note $x_i^- = \underline{x_i}$ and $x_i^+ = \overline{x_i}$ (resp. $x_i^- = \overline{x_i}$ and $x_i^+ = \underline{x_i}$). f_{min} and f_{max} are functions defined by:*

$$f_{min}(\mathcal{W}) = f(x_1^-, \dots, x_k^-, \mathcal{W})$$
$$f_{max}(\mathcal{W}) = f(x_1^+, \dots, x_k^+, \mathcal{W})$$

The monotonicity-based extension F_M of f computes:

$$F_M = [\underline{f_{min}(\mathcal{W})}, \overline{f_{max}(\mathcal{W})}]$$

Example 4. Let $f(x) = x^2 - 2x$ and $X = [1, 4]$. As seen in Example 1, $F_N([1, 4]) = [-7, 14]$. The derivative of f is $f'(x) = 2x - 2$, and $F'_N([1, 4]) = 2 \times [1, 4] - 2 = [0, 6] \geq 0$. f is thus increasing with respect to x in X. Therefore, the monotonicity-based interval extension computes the optimal range: $F_M([1, 4]) = [\underline{F(\underline{X})}, \overline{F(\overline{X})}] = [\underline{F(1)}, \overline{F(4)}] = [-1, 8] = f([1, 4])$.

This powerful property has been exploited in the contractor `Mohc` [10] and implemented in Charibde to enhance constraint propagation. However, the efficiency of this approach remains limited because the computation of partial derivatives may also be subject to overestimation.

4 Charibde Algorithm

We consider the following n-dimensional optimization problem and we assume that f is differentiable and that the analytical forms of f and its partial derivatives are available. We note n the dimension of the search-space.

$$\begin{aligned} \min_{x \in \mathbf{D} \subset \mathbb{R}^n} \quad & f(x) \\ \text{s.t.} \quad & g_i(x) \leq 0, \; i \in \{1, \dots, m\} \end{aligned}$$

The current work extends the core method described in [6], in which we combined a GA and an IB&B that ran independently, and cooperated by exchanging information through shared memory in order to accelerate the convergence. In this framework, the GA quickly finds satisfactory solutions that improve the upper bound $\tilde{f}$ of the global minimum, and allows the IB&B to prune parts of the search-space more efficiently.

The interval-based algorithm embedded in Charibde follows a **Branch & Contract** (IB&C) scheme (described in Algorithm 1), namely an IB&B algorithm that integrates a contraction step based on `HC4Revise`. An IB&B merely

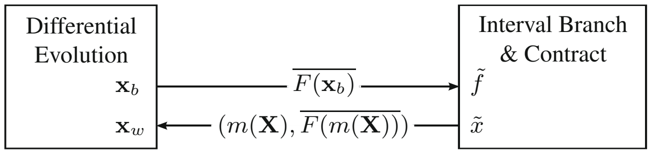

Fig. 3. Cooperative scheme of Charibde

relies on the refutation principle (discard a box if it is unfeasible or if it cannot contain a global minimizer). An IB&C may contract boxes by taking into account the constraints $g_i(x) \leq 0, i \in \{1, \ldots, m\}$ (feasibility) or $\frac{\partial f}{\partial x_i} = 0, i \in \{1, \ldots, n\}$ (local optimality) and $f \leq \tilde{f}$. Exploiting the analytical form of the objective function and its derivatives achieves faster convergence of the hybrid algorithm. Filtering algorithms show particular efficiency when $\tilde{f}$ is a good approximation of the global minimum provided by the EA thread, hence the necessity to quickly find an incumbent solution. Charibde thus outperforms our previous algorithm by far.

We note $\tilde{x}$ the best known solution, such that $\overline{F(\tilde{x})} = \tilde{f}$. The cooperation between the two threads boils down to three main steps (Fig. 3):

- Whenever the best known DE evaluation is improved, the best individual $\mathbf{x}_b$ is evaluated using IA. The upper bound of the image $\overline{F(\mathbf{x}_b)}$ – an upper bound of the global minimum – is sent to the IB&C thread
- In the IB&C algorithm, $\overline{F(\mathbf{x}_b)}$ is compared to the current best upper bound $\tilde{f}$. An improvement of the latter leads to a more efficiently pruning of the subspaces that cannot contain a (feasible) global minimizer
- Whenever the evaluation of the center $m(\mathbf{X})$ of a box improves $\tilde{f}$, the individual $m(\mathbf{X})$ replaces the worst individual $\mathbf{x}_w$ of DE, thus preventing premature convergence

In the following, we detail the implementations of the two main components of our algorithm: the deterministic IB&C thread and the stochastic DE thread.

4.1 Interval Branch & Contract Thread

We note $\mathcal{L}$ the priority queue in which the remaining boxes are stored and ε the desired precision. The basic framework of IB&C algorithms is described in Algorithm 1. The following rules have been experimentally tested and implemented in Charibde:

Selection rule: The box $\mathbf{X}$ for which $\overline{F(\mathbf{X})}$ is the largest is extracted from $\mathcal{L}$
Bounding rule: Evaluating $F(\mathbf{X})$ yields a rigorous enclosure of $f(\mathbf{X})$
Cut-off test: If $\tilde{f} - \varepsilon < \underline{F(\mathbf{X})}$, $\mathbf{X}$ is discarded as it cannot improve $\tilde{f}$ by more than ε
Midpoint test: If the evaluation of the midpoint of $\mathbf{X}$ improves $\tilde{f}$, $\tilde{f}$ is updated

Branching rule: $\mathbf{X}$ is bisected along the k-th dimension, where k is chosen according to the round-robin method (one dimension after another). The two resulting subboxes are inserted in $\mathcal{L}$ to be processed at a later stage

Algorithm 1. Interval Branch and Contract framework

```
f̃ ← +∞                                          ▷ best found upper bound
L ← {X0}                                         ▷ priority queue of boxes to process
repeat
    Extract a box X from L                       ▷ selection rule
    Compute F(X)                                 ▷ bounding rule
    if X cannot be eliminated then               ▷ cut-off test
        Contract(X, f̃)                           ▷ filtering algorithms
        Compute F(m(X)) to update f̃              ▷ midpoint test
        Bisect X into X1 and X2                  ▷ branching rule
        Store X1 and X2 in L
    end if
until L = ∅
return (f̃, x̃)
```

4.2 Differential Evolution Thread

Differential Evolution (DE) is an EA that combines the coordinates of existing individuals with a particular probability to generate new potential solutions [11]. It has shown great potential for solving difficult optimization problems, and has few control parameters. Let us denote NP the population size, $W > 0$ the weighting factor and $CR \in [0, 1]$ the crossover rate. For each individual $\mathbf{x}$ of the population, three other individuals $\mathbf{u}$, $\mathbf{v}$ and $\mathbf{w}$, all different and different from $\mathbf{x}$, are randomly picked in the population. The newly generated individual $\mathbf{y} = (y_1, \ldots, y_j, \ldots, y_n)$ is computed as follows:

$$y_j = \begin{cases} u_j + W \times (v_j - w_j) & \text{if } j = R \text{ or } rand(0,1) < CR \\ x_j & \text{otherwise} \end{cases} \tag{1}$$

R is a random index in $\{1, \ldots, n\}$ ensuring that at least one component of $\mathbf{y}$ differs from that of $\mathbf{x}$. $\mathbf{y}$ replaces $\mathbf{x}$ in the population if $f(\mathbf{y}) < f(\mathbf{x})$.

Boundary constraints: When a component y_j lies outside the bounds $[\underline{D_j}, \overline{D_j}]$ of the search-space, the *bounce-back method* [12] replaces y_j with a component that lies between u_j (the j-th component of $\mathbf{u}$) and the admissible bound:

$$y_j = \begin{cases} u_j + rand(0,1)(\overline{D_j} - u_j), & \text{if } y_j > \overline{D_j} \\ u_j + rand(0,1)(\underline{D_j} - u_j), & \text{if } y_j < \underline{D_j} \end{cases} \tag{2}$$

Evaluation: Given inequality constraints $\{g_i \mid i = 1, \ldots, m\}$, the evaluation of an individual $\mathbf{x}$ is computed as a triplet $(f_{\mathbf{x}}, n_{\mathbf{x}}, s_{\mathbf{x}})$, where $f_{\mathbf{x}}$ is the objective value, $n_{\mathbf{x}}$ the number of violated constraints and $s_{\mathbf{x}} = \sum_{i=1}^{m} \max(g_i(\mathbf{x}), 0)$. If at least one of the constraints is violated, the objective value is not computed

Selection: Given the evaluation triplets $(f_{\mathbf{x}}, n_{\mathbf{x}}, s_{\mathbf{x}})$ and $(f_{\mathbf{y}}, n_{\mathbf{y}}, s_{\mathbf{y}})$ of two candidate solutions $\mathbf{x}$ and $\mathbf{y}$, the best individual to be kept for the next generation is computed as follows:
- if $n_{\mathbf{x}} < n_{\mathbf{y}}$ or $(n_{\mathbf{x}} = n_{\mathbf{y}} > 0$ and $s_{\mathbf{x}} < s_{\mathbf{y}})$ or $(n_{\mathbf{x}} = n_{\mathbf{y}} = 0$ and $f_{\mathbf{x}} < f_{\mathbf{y}})$ then $\mathbf{x}$ is kept
- otherwise, $\mathbf{y}$ replaces $\mathbf{x}$

Termination: The DE has no termination criterion and stops only when the IB&C thread has reached convergence

5 Experimental Results

Charibde has been tested on a benchmark of standard test functions including quadratic, polynomial and nonlinear functions: bound-constrained problems (Rana, Egg Holder, Schwefel, Rosenbrock, Rastrigin, Michalewicz and Griewanlk) and inequality-constrained problems (Tension, Himmelblau, Welded Beam and Keane). Both the best known minimum in the literature and the certified global minimum[1] computed by Charibde are reported in Table 1. The global minima may be analytically computed for some separable or trivial functions, but for others (Rana and Egg Holder functions) no result concerning deterministic methods exists in the literature. Charibde has achieved new optimality results for three functions (Rana, Egg Holder and Michalewicz) and has proven the optimality of the known minima of the other functions.

Table 1. Test functions with best known and certified minima

	n	Type	Reference	Best known minimum	Certified minimum by Charibde
Bound-Constrained Problems					
Rana	4	Nonlinear	[15]	–	**−1535.1243381**
Egg Holder	10	Nonlinear	[16]	−8247 [17]	**−8291.2400675249**
Schwefel	10	Nonlinear		−4189.828873 [18]	−4189.8288727
Rosenbrock	50	Quadratic		0	0
Rastrigin	50	Nonlinear		0	0
Michalewicz	75	Nonlinear		–	**−74.6218111876**
Griewank	200	Nonlinear		0	0
Inequality-Constrained Problems					
Tension	3	Polynomial	[19]	0.012665232788319 [20]	0.0126652328
Himmelblau	5	Quadratic	[19]	−31025.560242 [21]	−31025.5602424972
Welded Beam	4	Nonlinear	[19]	1.724852309 [22]	1.7248523085974
Keane	5	Nonlinear	[23]	−0.634448687 [24]	−0.6344486869

Note that the constraints of Keane's function do not contain variables with multiple occurrences, and are therefore not subject to dependency. However, the first inequality constraint, describing a hyperbola in two dimensions, is active at

[1] Corresponding solutions are available upon request.

the global minimizer. The second inequality constraint is linear and is not active at the global minimizer. These constraints are highly combinatorial due to the sum and product operations, which makes constraint propagation rather inefficient. The Egg Holder (resp. Rana) function is strongly subject to dependency: x_1 and x_n occur three (resp. Five) times in its expression, and $(x_2, \dots, x_{n-1})$ occur six (resp. Ten) times. Their natural interval extensions therefore produce a large overestimation of the actual range. They are extremely difficult for interval-based solvers to optimize.

Partial derivatives of the objective function are computed using automatic differentiation [13]. To compute the partial derivatives of the functions that contain absolute values (Rana, Egg Holder, Schwefel and Keane), we use an interval extension based on the subderivative of $|\cdot|$ [14]:

$$|\cdot|'(X) = \begin{cases} [-1,-1] & \text{if } \overline{X} < 0 \\ [1,1] & \text{if } \underline{X} > 0 \\ [-1,1] & \text{otherwise} \end{cases} \quad (3)$$

The statistics of Charibde over 100 runs are presented in Table 2. ε is the numerical precision of the certified minimum such that $\tilde{f} - f^* < \varepsilon$, (NP, W, CR) are the DE parameters, t_{max} is the maximal computation time (in seconds), S_{max} is the maximal size of the priority queue $\mathcal{L}$, ne_f is the number of evaluations of the real-valued function f and $ne_F = ne_F^{DE} + ne_F^{IB\&C}$ is the number of evaluations of the interval function F computed in the DE thread (ne_F^{DE}) and the IB&C thread ($ne_F^{IB\&C}$). Note that ne_F^{DE} represents the number of improvements of the best DE evaluation. Because the DE thread keeps running as long as the IB&C thread has not achieved convergence, ne_f is generally much larger than

Table 2. Average results over 100 runs

	n	ε	NP	W	CR	t_{max}	S_{max}	ne_f	ne_F
Bound-Constrained Problems									
Rana	4	10^{-6}	50	0.7	0.5	222	42	274847000	$47 + 27771415$
Egg Holder	10	10^{-6}	50	0.7	0.5	768	45	423230200	$190 + 423230200$
Schwefel	10	10^{-6}	40	0.7	0.5	2.3	32	1462900	$150 + 362290$
Rosenbrock	50	10^{-12}	40	0.7	0.9	3.3	531	368028	$678 + 664914$
Rastrigin	50	10^{-15}	40	0.7	0	0.3	93	29372	$29 + 42879$
Michalewicz	75	10^{-9}	70	0.5	0	138	187	6053495	$1203 + 5796189$
Griewank	200	10^{-12}	50	0.5	0	11.8	134	188340	$316 + 116624$
Inequality-Constrained Problems									
Tension	3	10^{-9}	50	0.7	0.9	3.8	80	1324026	$113 + 1057964$
Himmelblau	5	10^{-9}	50	0.7	0.9	0.07	139	12147	$104 + 36669$
Beam	4	10^{-12}	50	0.7	0.9	2.2	11	316966	$166 + 54426$
Keane	5	10^{-4}	40	0.7	0.5	472	23	152402815	$125 + 99273548$

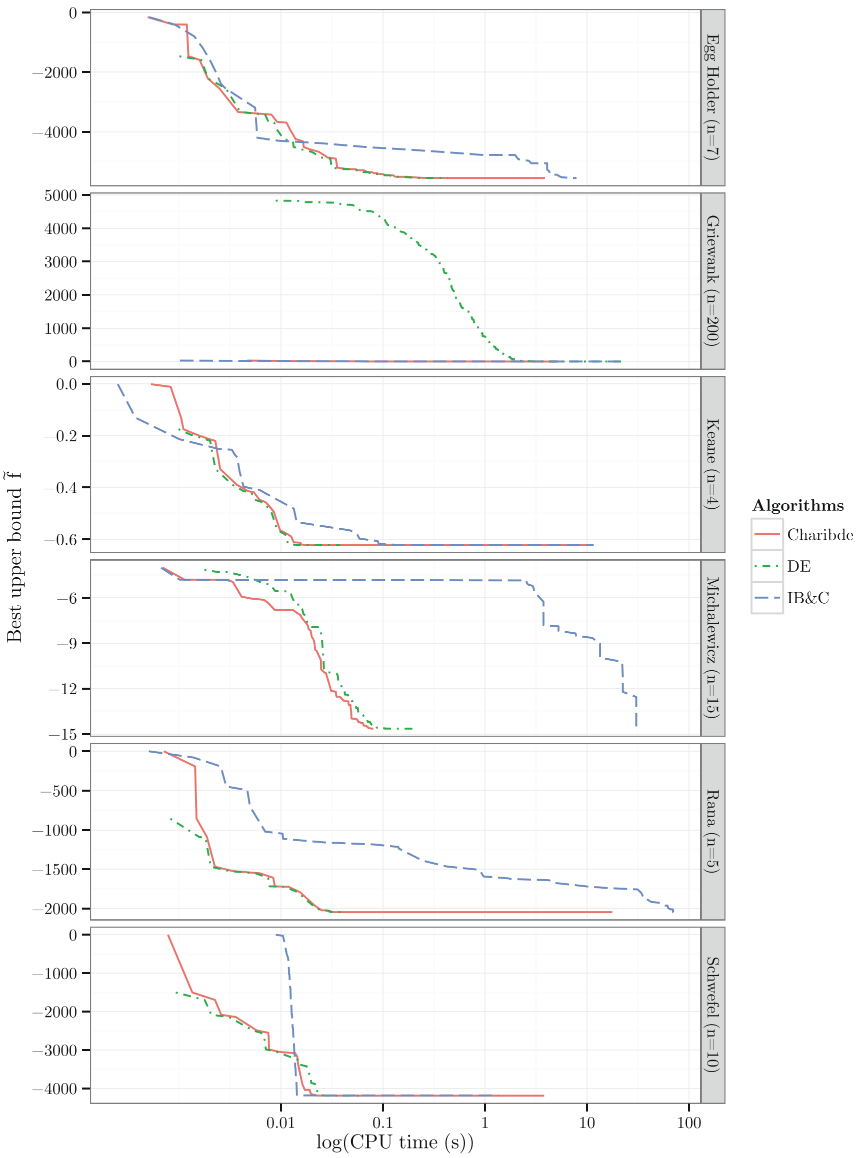

Fig. 4. Comparison of Charibde and standalone DE and IB&C (logarithmic x scale)

the number of evaluations required to reach $\tilde{f}$. These statistics suggest that the Egg Holder function, Keane's function and Rana's function are among the most challenging nonlinear problems for numerical solvers.

Figure 4 portrays the average comparison of performance between Charibde and standalone DE and IB&C over 100 runs of Six of the test functions (Egg Holder, Griewank, Keane, Michalewicz, Rana and Schwefel). A particular instance of each problem has been selected so that the standalone IB&C reaches convergence within reasonable time; to this end, the standard "best-first search" heuristic (extract the box $\mathbf{X}$ with the lowest $\underline{F(\mathbf{X})}$) seemed more suitable. The DE algorithm reaches the global minimum for all instances. The IB&C generally experiences several phases of stagnation: this is due to the (crude) upper bounds of f^* obtained when evaluating the center of the boxes. On the contrary, Charibde benefits from the start of convergence of either the DE algorithm (Egg Holder, Keane, Rana and Schwefel) or the IB&C algorithm (Griewank and Michalewicz) to reach the global minimum faster than its standalone methods. Charibde proves to be highly competitive with the IB&C algorithm: on these (relatively simple) instances, the gain ratios in CPU time are respectively 2.04 (Egg Holder), 3.93 (Griewank), 1.14 (Keane), 377 (Michalewicz) and 3.95 (Rana). The IB&C algorithm however turns out to be more efficient than Charibde on the Schwefel function (gain ratio in CPU time: 0.36).

6 Conclusion

Extending the basic concept of [6], we have presented in this paper a new cooperative hybrid algorithm, Charibde, in which a stochastic Differential Evolution algorithm (DE) cooperates with a deterministic Interval Branch and Contract algorithm (IB&C). The DE algorithm quickly finds incumbent solutions that help the IB&C to improve pruning the search-space using interval propagation techniques. Whenever the IB&C improves the best known upper bound $\tilde{f}$ of the global minimum f^*, the corresponding solution is used as a new DE individual to avoid premature convergence toward local optima.

We have demonstrated the efficiency of this algorithm on a benchmark of difficult multimodal functions. Previously unknown results have been presented for Rana, Egg Holder and Michalewicz functions, while other known minima have been certified. By preventing premature convergence in the DE algorithm and providing the IB&C algorithm with a good approximation $\tilde{f}$ of f^*, Charibde proves highly competitive with its two standalone components.

References

1. Moore, R.E.: Interval Analysis. Prentice-Hall, Englewood Cliffs (1966)
2. Sotiropoulos, G.D., Stavropoulos, C.E., Vrahatis, N.M.: A new hybrid genetic algorithm for global optimization. In: Proceedings of Second World Congress on Nonlinear Analysts, pp. 4529–4538. Elsevier Science Publishers Ltd. (1997)
3. Holland, J.H.: Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor (1975)
4. Zhang, X., Liu, S.: A new interval-genetic algorithm. Int. Conf. Nat. Comput. **4**, 193–197 (2007)
5. Lei, Y., Chen, S.: A reliable parallel interval global optimization algorithm based on mind evolutionary computation. In: 2012 Seventh ChinaGrid Annual Conference, pp. 205–209 (2009)
6. Alliot, J.M., Durand, N., Gianazza, D., Gotteland, J.B.: Finding and proving the optimum: cooperative stochastic and deterministic search. In: 20th European Conference on Artificial Intelligence (2012)

7. Hansen, E.: Global Optimization Using Interval Analysis. Dekker, New York (1992)
8. Chabert, G., Jaulin, L.: Contractor programming. Artif. Intell. **173**, 1079–1100 (2009)
9. Benhamou, F., Goualard, F., Granvilliers, L., Puget, J.F.: Revising hull and box consistency. In: International Conference on Logic Programming, pp. 230–244. MIT press, Cambridge (1999)
10. Araya, I., Trombettoni, G., Neveu, B.: Exploiting monotonicity in interval constraint propagation. In: Proceedings of the AAAI, pp. 9–14 (2010)
11. Storn, R., Price, K.: Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. J. Global Optim. **11**, 341–359 (1997)
12. Price, K., Storn, R., Lampinen, J.: Differential Evolution - A Practical Approach to Global Optimization. Natural Computing. Springer, New York (2006)
13. Rall, L.B.: Automatic Differentiation: Techniques and Applications. LNCS, vol. 120. Springer, Heidelberg (1981)
14. Kearfott, R.B.: Interval extensions of non-smooth functions for global optimization and nonlinear systems solvers. Computing **57**, 57–149 (1996)
15. Whitley, D., Mathias, K., Rana, S., Dzubera, J.: Evaluating evolutionary algorithms. Artif. Intell. **85**, 245–276 (1996)
16. Mishra, S.K.: Some new test functions for global optimization and performance of repulsive particle swarm method. Technical report, University Library of Munich, Germany (2006)
17. Sekaj, I.: Robust parallel genetic algorithms with re-initialisation. In: Yao, X., et al. (eds.) PPSN 2004. LNCS, vol. 3242, pp. 411–419. Springer, Heidelberg (2004)
18. Kim, Y.H., Lee, K.H., Yoon, Y.: Visualizing the search process of particle swarm optimization. In: Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation, pp. 49–56. ACM (2009)
19. Coello Coello, C.A.: Use of a self-adaptive penalty approach for engineering optimization problems. Comput. Ind. **41**, 113–127 (1999)
20. Zhang, J., Zhou, Y., Deng, H.: Hybridizing particle swarm optimization with differential evolution based on feasibility rules. In: ICGIP 2012, vol. 8768 (2013)
21. Aguirre, A., Muñoz Zavala, A., Villa Diharce, E., Botello Rionda, S.: COPSO: constrained optimization via PSO algorithm. Technical report, CIMAT (2007)
22. Duenez-Guzman, E., Aguirre, A.: The Baldwin effect as an optimization strategy. Technical report, CIMAT (2007)
23. Keane, A.J.: A brief comparison of some evolutionary optimization methods. In: Proceedings of the Conference on Applied Decision Technologies. Modern Heuristic Search Methods, Uxbridge, 1995, pp. 255–272. Wiley, Chichester (1996)
24. Mishra, S.K.: Minimization of Keane‘s bump function by the repulsive particle swarm and the differential evolution methods. Technical report, North-Eastern Hill University, Shillong (India) (2007)